Project Two Report

Pooya Fayyazsanavi
Computer Science
George Mason University
Virginia, USA
pfayyazs@gmu.edu

***Abstract*—This is a report for project two.**

I. Name on miner and F1-score

My submission name on miner : **Mr.Nobody**
Submission F1-score : 0.69

II. Name on miner and F1-score

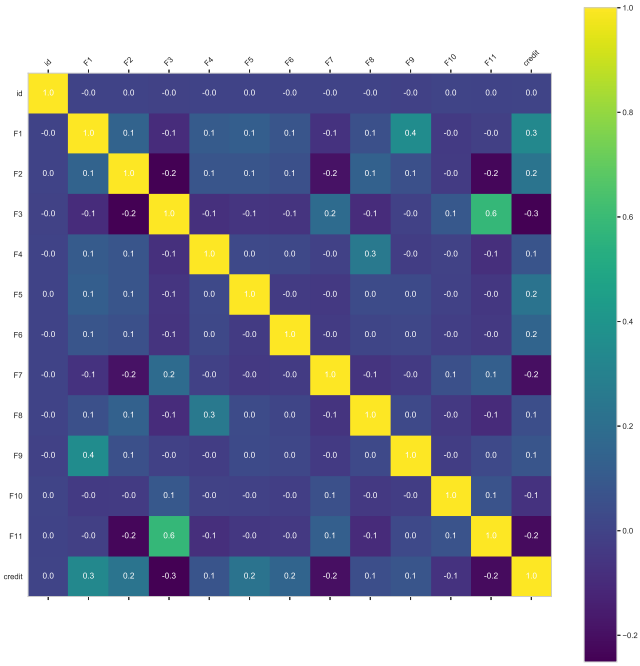

Fig. 1. Correlation between the features

III. Approach

In this section the approach to final results are discussed thoroughly.

A. Reading the data

At first the data converted to pandas data frame with *read_csv* function. Pandas data frame is a great tool which we can filter and apply functions to our data easily.

B. Plotting the data Correlation

One of the good ways to know how your features are related to each other is to plot the correlation matrix. For instance, figure 2 shows that there is no correlation between feature *ID* and the label which is *Credit*. For this task I just removed the feature ID and other features remain untouched.

C. Categorical to Numerical

If we want fit our data to a model, we have to convert the categorical features to numerical representation. There are two columns which we have to convert, the *Race* and *Gender* features. Simply, I use numerical values from one for each unique value. For example, I map "Male" to 0 and "Female" to 1.

IV. Approach to solve imbalance Data-set

Within the training set there are 24720 samples with label (0) and 7841 samples with label (1). In order to get a generalized model we have to balance our data-set. I tried two different ways to solve this problem.

A. Over-sampling

one of the easiest ways to solve this problem would be generating synthetic data points. These data can be generated by duplicating rows or taking mean of n samples or using the K-NN algorithm to generate new row. I used *SMOTE* library to generate my synthetic data points. Although, the results were improved in the K-Fold validation, there was not any improvement in the test results.

B. Under-sampling

The other way to come up with a solution to this problem would be reducing the samples until the diversity is equal for all classes. This can be done by removing the rows in class label 0 until the rows become equal to class 1. For this section I used the $RandomUnderSampler$. The results for the test data-set improved by a good factor.

V. Data Pre-processing

A. Normalizing the data

Two of the features have a high value. The values of gains and loss features are numerically large and it could effect on the results and it also makes the algorithm run longer. One the best approaches would be scaling all of the features. I used the $StandardScaler$ which maps all of the values with the formula below.

$$z = (x - u)/s$$

where u is the mean of the training samples, and s is the standard deviation of the training samples.

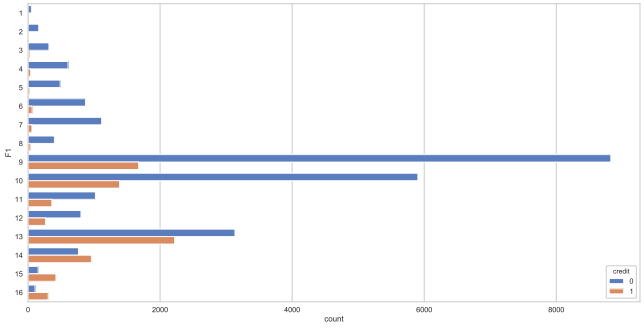

Fig. 2. relation between F1 and Credit feature

B. PCA

I also used the PCA feature reduction technique. I used the PCA to map my original data and reduce the features to 3. The PCA was not help full in this part and it was not able to improve the F1-score.

VI. Models

To get the best results I tested five different models with the under sampled data. The algorithms are as follows.

A. Logistic Regression model

The logistic model is used to model the probability of a certain class. In this case the model tries to predict our final label. Logistic Regression model is a statistical model. The outcome of the model would be binary. The results for this model was 0.751 for the validation set Score.

B. Random Forest Classifier

The Random Forest Classifier is built with ensemble multiple trees together. The logic behind it is somehow the same as decision tree. However, it takes the majority of votes for generating the final label. I was able to get 0.81 for the validation set Score.

C. XGB Classifier

XGB Classifier is also a tree designed classifier. The classifier itself was really fast and efficient. I got the best results with this classifier. The results on validation set was 0.83 Score.

D. SVM Classifier

SVM is one the best algorithms in the classification area. It is robust to outliers and also you can use kernels to transform the data into another dimensionality. The results on validation set was 0.80 Score.

E. MLP

Multi layer perceptron was the last classifier that I used. It uses the weights and bias's and keep updating them during the training time. I used four hidden layer with $(40, 10, 5, 3)$ neurons in each layer. The results on validation set was 0.80 Score.

VII. F-Fold and Hyper-Parameter tuning with $GridSearchCV$

All of these models mentioned above, they have multiple hyper-parameters which need to set. $GridSearchCV$ Is a great way to evaluate your model with different parameters. I tried different parameters for every model and got the best parameters for the model. Also, in order to evaluate the models I used the K-Fold cross validation to get the best model. The $XGBClassifier$ was the best model which outperforms the others. I got the 0.69 on the miner with this model.